Package 'EnrichedHeatmap'

August 17, 2018

Type Package

Title Making Enriched Heatmaps

Version 1.10.0

Date 2018-4-6

Author Zuguang Gu

Maintainer Zuguang Gu <z.gu@dkfz.de>

Depends R (>= 3.1.2), methods, grid, ComplexHeatmap (>= 1.15.2), GenomicRanges

Imports matrixStats, stats, GetoptLong, Rcpp, utils, locfit, circlize (>= 0.4.1), IRanges

Suggests testthat (>= 0.3), knitr, markdown, genefilter, RColorBrewer

VignetteBuilder knitr

Description Enriched heatmap is a special type of heatmap which visualizes the enrichment of genomic signals on specific target regions. Here we implement enriched heatmap by ComplexHeatmap package. Since this type of heatmap is just a normal heatmap but with some special settings, with the functionality of ComplexHeatmap, it would be much easier to customize the heatmap as well as concatenating to a list of heatmaps to show correspondance between different data sources.

biocViews Software, Visualization, Sequencing, GenomeAnnotation, Coverage

URL https://github.com/jokergoo/EnrichedHeatmap

License MIT + file LICENSE

Repository Bioconductor

LinkingTo Rcpp

git_url https://git.bioconductor.org/packages/EnrichedHeatmap

git_branch RELEASE_3_7

git_last_commit 1aafeb1

git_last_commit_date 2018-04-30

Date/Publication 2018-08-16

R topics documented:

+.AdditiveUnit . . . 2
anno_enriched . . . 3
copyAttr . . . 4
default_smooth_fun . . . 5
discretize . . . 6
dist_by_closeness . . . 7
draw-dispatch . . . 7
draw-EnrichedHeatmap-method . . . 8
draw-EnrichedHeatmapList-method . . . 9
EnrichedHeatmap . . . 10
EnrichedHeatmap-class . . . 11
EnrichedHeatmapList . . . 12
EnrichedHeatmapList-class . . . 13
enriched_score . . . 13
extract_anno_enriched . . . 14
getSignalsFromList . . . 15
makeWindows . . . 16
normalizeToMatrix . . . 17
print.normalizedMatrix . . . 19
rbind.normalizedMatrix . . . 20
show-dispatch . . . 21
show-EnrichedHeatmap-method . . . 21
show-EnrichedHeatmapList-method . . . 22
[.normalizedMatrix . . . 22

Index **24**

`+.AdditiveUnit` *Add heatmaps or row annotations to a heatmap list*

Description

Add heatmaps or row annotations to a heatmap list

Usage

```
## S3 method for class 'AdditiveUnit'
x + y
```

Arguments

x	an `EnrichedHeatmap-class` object, a `Heatmap-class` object, a `HeatmapAnnotation-class` object or a `HeatmapList-class` object.
y	an `EnrichedHeatmap-class` object, a `Heatmap-class` object, a `HeatmapAnnotation-class` object or a `HeatmapList-class` object.

Details

It overwrites `+.AdditiveUnit` in the ComplexHeatmap package.

Value

A HeatmapList-class object or an EnrichedHeatmapList-class object

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
# users should not use it directly
NULL
```

---

anno_enriched *Annotation function to show the enrichment*

Description

Annotation function to show the enrichment

Usage

```
anno_enriched(gp = gpar(col = "red"), pos_line = TRUE, pos_line_gp = gpar(lty = 2),
    yaxis = TRUE, ylim = NULL, value = c("mean", "sum", "abs_mean", "abs_sum"),
   yaxis_side = "right", yaxis_facing = ifelse(yaxis_side == "right", "right", "left"),
    yaxis_gp = gpar(fontsize = 8), show_error = FALSE)
```

Arguments

gp	graphic parameters. There are two non-standard parameters: neg_col and pos_col. If these two parameters are defined, the positive signals and negatie signals are visualized separatedly. The graphic parameters can be set as vectors when the heatmap or heatmap list is split into several row clusters.
pos_line	whether to draw vertical lines which represent positions of target
pos_line_gp	graphic parameters for the position lines
yaxis	whether show yaxis
ylim	ranges on y-axis, by default it is inferred from the data
value	the method to summarize signals from columns of the noramlized matrix
yaxis_side	side of y-axis
yaxis_facing	facing of the axis ticks and labels. It can be set to avoid overlapping text when multiple heatmaps are plotted together
yaxis_gp	graphic parameters for y-axis
show_error	whether show error regions which are one standard error to the mean value. Color of error area is same as the corresponding lines with 75 percent transparency.

Details

This annotation functions shows mean values (or depends on the method set in `value` argument) of columns in the normalized matrix which summarises the enrichment of the signals to the targets.

If rows are splitted, the enriched lines are calculated for each row cluster and there will also be multiple lines in this annotation viewport.

It should only be placed as column annotation of the enriched heatmap.

Value

A column annotation function which can be set to `top_annotation` argument in `EnrichedHeatmap`.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
load(system.file("extdata", "chr21_test_data.RData", package = "EnrichedHeatmap"))
tss = promoters(genes, upstream = 0, downstream = 1)
mat1 = normalizeToMatrix(H3K4me3, tss, value_column = "coverage",
    extend = 5000, mean_mode = "w0", w = 50, keep = c(0, 0.99))
EnrichedHeatmap(mat1, col = c("white", "red"), name = "H3K4me3",
    top_annotation = HeatmapAnnotation(lines = anno_enriched(gp = gpar(col = 2:4))),
    km = 3, row_title_rot = 0)
```

---

copyAttr *Copy attributes to another object*

Description

Copy attributes to another object

Usage

```
copyAttr(x, y)
```

Arguments

x	object 1
y	object 2

Details

The `normalizeToMatrix` object is actually a matrix but with more additional attributes attached. When manipulating such matrix, there are some circumstances that the attributes are lost. This function is used to copy these specific attributes when dealing with the matrix.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
gr = GRanges(seqnames = c("chr5", "chr5"),
ranges = IRanges(start = c(98, 98),
                 end = c(104, 104)))
target = GRanges(seqnames = "chr5",
ranges = IRanges(start = 100,
             end = 100))
mat1 = normalizeToMatrix(gr, target, extend = 6, w = 1)
# attributes removed and you cannot use it for EnrichedHeatmap()
mat2 = mat1[]
# copy attributes to mat2 and now mat3 can be used for EnrichedHeatmap()
mat3 = copyAttr(mat1, mat2)
```

`default_smooth_fun` *Default smoothing function*

Description

Default smoothing function

Usage

```
default_smooth_fun(x)
```

Arguments

x	input numeric vector

Details

The smoothing function is applied to every row in the normalized matrix. For this default smoothing function, `locfit` is first tried on the vector. If there is error, `loess` smoothing is tried afterwards. If both smoothing are failed, there will be an error.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
# There is no example
NULL
```

discretize *Discretize a continuous matrix to a discrete matrix*

Description

Discretize a continuous matrix to a discrete matrix

Usage

```
discretize(mat, rule, right_closed = FALSE)
```

Arguments

mat	a normalize matrix from normalizeToMatrix.
rule	a list of intervals which provide mapping between continuous values to discrete values. Note the order of intervals determines the order of corresponding discrete levels.
right_closed	is the interval right closed?

Details

Assuming we have a normalized matrix with both positive values and negative values, we only want to see the enrichment of the windows/regions showing significant positive values and negative values and we are only interested in the direction of the values while not the value itself, then we can define the rule as:

```
rule = list(
    "positive" = c(0.5, Inf),
    "negative" = c(-Inf, -0.5)
)
```

And we can convert the continuous matrix to a discrete matrix and visualize it:

```
mat2 = discretize(mat, rule)
EnrichedHeatmap(mat2, col = c("positive" = "red", "negative" = "green"))
```

Another example is to discretize the signals to discrete levels according to the intensities:

```
rule = list(
    "very_high" = c(100, Inf),
    "high" = c(50, 100),
    "intermediate" = c(25, 50),
    "low" = c(1e-6, 25)
)
```

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
# There is no example
NULL
```

dist_by_closeness *Distance by closeness*

Description

Distance by closeness

Usage

```
dist_by_closeness(mat)
```

Arguments

mat	a numeric matrix where the distance is calculated by rows

Details

For two rows in the matrix, assume x_1, x_2, ..., x_n1 are the column index of none-zero values in row 1 and y_1, y_2, ... y_n2 are the column index for non-zero values in row 2, the distance between the two rows based on the closeness is calculated as:

d_closeness = sum_i sum_j(|x_i - y_j|) / (n_1*n_2)

Value

A dist object

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
x1 = c(0, 0, 0, 0, 1, 1, 1, 0, 0, 0)
x2 = c(0, 0, 0, 1, 1, 1, 0, 0, 0, 0)
x3 = c(1, 0, 0, 0, 1, 1, 0, 0, 0, 0)
m = rbind(x1, x2, x3)
dist(m)
dist_by_closeness(m)
```

draw-dispatch *Method dispatch page for draw*

Description

Method dispatch page for draw.

Dispatch

draw can be dispatched on following classes:

- draw,EnrichedHeatmapList-method, EnrichedHeatmapList-class class method
- draw,EnrichedHeatmap-method, EnrichedHeatmap-class class method

Examples

```
# no example
NULL
```

`draw-EnrichedHeatmap-method`
Draw a single heatmap

Description

Draw a single heatmap

Usage

```
## S4 method for signature 'EnrichedHeatmap'
draw(object, internal = FALSE, ...)
```

Arguments

`object`	an `EnrichedHeatmap-class` object.
`internal`	only used internally.
`...`	pass to `draw`,HeatmapList-method.

Details

The function creates an `EnrichedHeatmapList-class` object which only contains a single heatmap and call `draw,EnrichedHeatmapList-method` to make the final heatmap.

Value

An `EnrichedHeatmapList-class` object.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
# see documentation of EnrichedHeatmap
NULL
```

draw-EnrichedHeatmapList-method
Draw a list of heatmaps

Description

Draw a list of heatmaps

Usage

```
## S4 method for signature 'EnrichedHeatmapList'
draw(object, padding = unit(c(2, 2, 2, 2), "mm"),
    newpage= TRUE, ...)
```

Arguments

object	an EnrichedHeatmapList-class object
padding	padding of the plot. The four values correspond to bottom, left, top, right paddings.
newpage	whether to create a new page
...	pass to make_layout, HeatmapList-method or draw, HeatmapList-method

Details

It calls draw, HeatmapList-method to make the plot but with some adjustment specificly for enriched heatmaps.

Value

An EnrichedHeatmapList object

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
# see documentation of EnrichedHeatmap
NULL
```

EnrichedHeatmap *Constructor method for EnrichedHeatmap class*

Description

Constructor method for EnrichedHeatmap class

Usage

```
EnrichedHeatmap(mat, col, top_annotation = HeatmapAnnotation(enriched = anno_enriched()),
    top_annotation_height = unit(2, "cm"),
    row_order = order(enriched_score(mat), decreasing = TRUE), pos_line = TRUE,
    pos_line_gp = gpar(lty = 2), axis_name = NULL, axis_name_rot = 0,
    axis_name_gp = gpar(fontsize = 10), border = TRUE, cluster_rows = FALSE,
    row_dend_reorder = -enriched_score(mat),
    show_row_dend = FALSE, show_row_names = FALSE,
    heatmap_legend_param = list(), ...)
```

Arguments

mat	a matrix which is returned by normalizeToMatrix
col	color settings. If the signals are categorical, color should be a vector with category levels as names.
top_annotation	a specific annotation which is always put on top of the enriched heatmap and is constructed by anno_enriched
top_annotation_height	the height of the top annotation
row_order	row order. Default rows are ordered by enriched scores calculated from enriched_score
pos_line	whether draw vertical lines which represent the positions of target
pos_line_gp	graphic parameters for the position lines
axis_name	names for axis which is below the heatmap. If the targets are single points, axis_name is a vector of length three which corresponds to upstream, target itself and downstream. If the targets are regions with width larger than 1, axis_name should be a vector of length four which corresponds to upstream, start of targets, end of targets and downstream.
axis_name_rot	rotation for axis names
axis_name_gp	graphic parameters for axis names
border	whether show border of the heatmap
cluster_rows	clustering on rows are turned off by default
show_row_dend	whether show dendrograms on rows if apply hierarchical clustering on rows
row_dend_reorder	weight for reordering the row dendrogram. It is reordered by enriched scores by default.
show_row_names	whether show row names
heatmap_legend_param	a list of settings for heatmap legends. at and labels can not be set here.
...	pass to Heatmap

Details

EnrichedHeatmap-class is inherited from Heatmap-class. Following parameters are set with pre-defined values:

`cluster_columns` enforced to be `FALSE`

`show_column_names` enforced to be `FALSE`

`bottom_annotation` enforced to be `NULL`

`column_title_side` enforced to be `top`

A EnrichedHeatmap-class object is also a Heatmap-class object, thus, most of the arguments in Heatmap are usable in EnrichedHeatmap such as to apply clustering on rows, or to split rows by data frame or k-means clustering. Users can also add more than one heatmaps by + operator. For a detailed demonstration, please go to the vignette.

Value

An EnrichedHeatmap-class object which is inherited from Heatmap-class.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
load(system.file("extdata", "chr21_test_data.RData", package = "EnrichedHeatmap"))
mat3 = normalizeToMatrix(meth, cgi, value_column = "meth", mean_mode = "absolute",
    extend = 5000, w = 50, smooth = TRUE)
EnrichedHeatmap(mat3, name = "methylation", column_title = "methylation near CGI")
EnrichedHeatmap(mat3, name = "meth1") + EnrichedHeatmap(mat3, name = "meth2")
# for more examples, please go to the vignette
```

EnrichedHeatmap-class *Class for a single heatmap*

Description

Class for a single heatmap

Details

The EnrichedHeatmap-class is inherited from Heatmap-class.

Methods

The EnrichedHeatmap-class provides following methods:

- EnrichedHeatmap: constructor method.
- draw,EnrichedHeatmap-method: draw a single heatmap.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

See Also

EnrichedHeatmapList-class

Examples

```
# There is no example
NULL
```

EnrichedHeatmapList *Constructor method for EnrichedHeatmapList class*

Description

Constructor method for EnrichedHeatmapList class

Usage

```
EnrichedHeatmapList(...)
```

Arguments

...	arguments

Details

There is no public constructor method for the EnrichedHeatmapList-class.

Value

No value is returned.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
# no example
NULL
```

EnrichedHeatmapList-class *Class for a list of heatmaps*

Description

Class for a list of heatmaps

Details

The EnrichedHeatmapList-class is inherited from HeatmapList-class.

Methods

The EnrichedHeatmapList-class provides following methods:

- draw,EnrichedHeatmapList-method: draw a list of heatmaps.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
# There is no example
NULL
```

enriched_score *Enriched scores*

Description

Enriched scores

Usage

```
enriched_score(mat)
```

Arguments

mat	a normalized matrix from normalizeToMatrix

Details

The function calculates how the signal is enriched in the target by weighting the distance to the target.

For a numeric vector, assume the vector is denoted as combination of three sub-vectors `c(x1, x2, x3)` with length `n1`, `n2` and `n3`, where `x1` are data points in upstream windows, `x2` are data points in target windows and `x3` are data points in downstream windows, the enriched score is calcualted as

sum(x_1i* i/n1) + sum(x_3j* (n3 - j + 1)/n3) + sum(x_2k * abs(n2/2 - abs(k - n2/2)))

where the first two terms are the distance to the start or end position of the target by weighting the distance to the position that if it is closer to the start or end position of the target, it has higher weight. The second term weight the distance to the center point of the target and similar, if it is closer to the center position, it has higher weight.

Value

A numeric vector

Author(s)

Zuguang Gu <z.gu@dkfz.de>

See Also

This `enriched_score` is the default scoring function for `score_fun` argument in `EnrichedHeatmap` function. It is also an example function for implementing customized scoreing function. Basically, to be a score function which calculates enriched score, it should accept three arguments which are the values in upstream windows, the target windows and downstream windows The user-defined function should return a single value. Rows are sorted decreasingly by the enriched scores.

Examples

```
# There is no example
NULL
```

---

extract_anno_enriched *Extarct enrichment annotation graphic as a separate plot*

Description

Extarct enrichment annotation graphic as a separate plot

Usage

```
extract_anno_enriched(ht_list, which = NULL, newpage = TRUE)
```

Arguments

`ht_list`	the heatmap list returned by `draw,EnrichedHeatmapList-method`
`which`	the index of enriched heamtap in the heatmap list. The value can be an integer index or a character index (which are names of heatmaps)
`newpage`	whether call `grid.newpage` to create a new page

Details

The extracted plot is exactly the same as that on the heatmap.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
# There is no example
NULL
```

`getSignalsFromList` *Get signals from a list*

Description

Get signals from a list

Usage

```
getSignalsFromList(lt, fun = function(x) mean(x, na.rm = TRUE))
```

Arguments

`lt`	a list of normalized matrices which are returned by `normalizeToMatrix`. Matrices in the list should be generated with same settings (e.g. they should use same target regions, same extension to targets and same number of windows).
`fun`	a user-defined function to summarize signals.

Details

Let's assume you have a list of histone modification signals for different samples and you want to visualize the mean pattern across samples. You can first normalize histone mark signals for each sample and then calculate means values across all samples. In following example code, `hm_gr_list` is a list of `GRanges` objects which contain positions of histone modifications, `tss` is a `GRanges` object containing positions of gene TSS.

```
mat_list = NULL
for(i in seq_along(hm_gr_list)) {
  mat_list[[i]] = normalizeToMatrix(hm_gr_list[[i]], tss, value_column = "density")
}
```

If we compress the list of matrices as a three-dimension array where the first dimension corresponds to genes, the second dimension corresponds to windows and the third dimension corresponds to samples, the mean signal across all sample can be calculated on the third dimension. Here `getSignalsFromList` simplifies this job.

Applying `getSignalsFromList()` to `mat_list`, it gives a new normalized matrix which contains mean signals across all samples and can be directly used in `EnrichedHeatmap()`.

```
mat_mean = getSignalsFromList(mat_list)
EnrichedHeatmap(mat_mean)
```

The correlation between histone modification and gene expression can also be calculated on the third dimension of the array. In the user-defined function `fun`, `x` is the vector for gene i and window j in the array, and `i` is the index of current gene.

```
mat_corr = getSignalsFromList(mat_list,
    fun = function(x, i) cor(x, expr[i, ], method = "spearman"))
```

Then `mat_corr` here can be used to visualize how gene expression is correlated to histone modification around TSS.

```
EnrichedHeatmap(mat_corr)
```

Value

A `normalizeToMatrix` object which can be directly used for `EnrichedHeatmap`.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
NULL
```

`makeWindows` *Split regions into windows*

Description

Split regions into windows

Usage

```
makeWindows(query, w = NULL, k = NULL, direction = c("normal", "reverse"),
    short.keep = FALSE)
```

Arguments

`query`	a `GRanges-class` object.
`w`	window size, a value larger than 1 means the number of base pairs and a value between 0 and 1 is the percent to the current region.
`k`	number of partitions for each region. If it is set, all other arguments are ignored.
`direction`	where to start the splitting. See 'Details' section.
`short.keep`	if the the region can not be split equally under the window size, the argument controls whether to keep the windows that are smaller than the window size. See 'Details' section.

Details

Following illustrates the meaning of `direction` and `short.keep`:

```
-->-->-->-  one region, split by 3bp window (">" represents the direction of the sequence)
 aaabbbccc   direction = "normal",  short.keep = FALSE
 aaabbbcccd  direction = "normal",  short.keep = TRUE
  aaabbbccc  direction = "reverse", short.keep = FALSE
 abbbcccddd  direction = "reverse", short.keep = TRUE
```

Value

A GRanges-class object with two additional columns attached:

- `.i_query` which contains the correspondance between small windows and original regions in `query`
- `.i_window` which contains the index of the small window on the current region.

Author(s)

Zuguang gu <z.gu@dkfz.de>

Examples

```
query = GRanges(seqnames = "chr1", ranges = IRanges(start = c(1, 11, 21), end = c(10, 20, 30)))
makeWindows(query, w = 2)
makeWindows(query, w = 0.5)
makeWindows(query, w = 3)
makeWindows(query, w = 3, direction = "reverse")
makeWindows(query, w = 3, short.keep = TRUE)
makeWindows(query, w = 3, direction = "reverse", short.keep = TRUE)
makeWindows(query, w = 12)
makeWindows(query, w = 12, short.keep = TRUE)
makeWindows(query, k = 2)
makeWindows(query, k = 3)
query = GRanges(seqnames = "chr1", ranges = IRanges(start = c(1, 11, 31), end = c(10, 30, 70)))
makeWindows(query, w = 2)
makeWindows(query, w = 0.2)
```

---

normalizeToMatrix *Normalize associations between genomic signals and target regions into a matrix*

Description

Normalize associations between genomic signals and target regions into a matrix

Usage

```
normalizeToMatrix(signal, target, extend = 5000, w = max(extend)/50,
    value_column = NULL, mapping_column = NULL, background = ifelse(smooth, NA, 0), empty_value = NU
    mean_mode = c("absolute", "weighted", "w0", "coverage"), include_target = any(width(target) > 1
    target_ratio = min(c(0.4, mean(width(target))/(sum(extend) + mean(width(target))))),
    k = min(c(20, min(width(target)))), smooth = FALSE, smooth_fun = default_smooth_fun,
     keep = c(0, 1), trim = NULL)
```

Arguments

signal	a GRanges-class object.
target	a GRanges-class object.
extend	extended base pairs to the upstream and/or downstream of target. It can be a vector of length one or two. Length one means same extension to the upstream and downstream.
w	window size for splitting upstream and downstream, measured in base pairs
value_column	column index in signal that is mapped to colors. If it is not set, it assumes values for all signal regions are 1.
mapping_column	mapping column to restrict overlapping between signal and target. By default it tries to look for all regions in signal that overlap with every target.
background	values for windows that don't overlap with signal.
empty_value	deprecated, please use background instead.
mean_mode	when a window is not perfectly overlapped to signal, how to summarize values to the window. See 'Details' section for a detailed explanation.
include_target	whether include target in the heatmap. If the width of all regions in target is 1, include_target is enforced to FALSE.
target_ratio	the ratio of target columns in the normalized matrix. If the value is 1, extend will be reset to 0.
k	number of windows only when target_ratio = 1 or extend == 0, otherwise ignored.
smooth	whether apply smoothing on rows in the matrix.
smooth_fun	the smoothing function that is applied to each row in the matrix. This self-defined function accepts a numeric vector (may contain NA values) and returns a vector with same length. If the smoothing is failed, the function should call stop to throw errors so that normalizeToMatrix can catch how many rows are failed in smoothing. See the default default_smooth_fun for example.
keep	percentiles in the normalized matrix to keep. The value is a vector of two percent values. Values less than the first percentile is replaces with the first pencentile and values larger than the second percentile is replaced with the second percentile.
trim	deprecated, please use keep instead.

Details

In order to visualize associations between signal and target, the data is transformed into a matrix and visualized as a heatmap by EnrichedHeatmap afterwards.

Upstream and downstream also with the target body are splitted into a list of small windows and overlap to signal. Since regions in signal and small windows do not always 100 percent overlap, there are four different averaging modes:

Following illustrates different settings for mean_mode (note there is one signal region overlapping with other signals):

```
  40      50     20     values in signal regions
++++++   +++    +++++   signal regions
       30               values in signal region
     ++++++             signal region
```

```
 =================    a window (17bp), there are 4bp not overlapping to any signal regions.
    4  6  3      3      overlap

absolute: (40 + 30 + 50 + 20)/4
weighted: (40*4 + 30*6 + 50*3 + 20*3)/(4 + 6 + 3 + 3)
w0:       (40*4 + 30*6 + 50*3 + 20*3)/(4 + 6 + 3 + 3 + 4)
coverage: (40*4 + 30*6 + 50*3 + 20*3)/17
```

Value

A matrix with following additional attributes:

`upstream_index` column index corresponding to upstream of `target`

`target_index` column index corresponding to `target`

`downstream_index` column index corresponding to downstream of `target`

`extend` extension on upstream and downstream

`smooth` whether smoothing was applied on the matrix

`failed_rows` index of rows which are failed after smoothing

The matrix is wrapped into a simple `normalizeToMatrix` class.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
signal = GRanges(seqnames = "chr1",
    ranges = IRanges(start = c(1, 4, 7, 11, 14, 17, 21, 24, 27),
                     end = c(2, 5, 8, 12, 15, 18, 22, 25, 28)),
    score = c(1, 2, 3, 1, 2, 3, 1, 2, 3))
target = GRanges(seqnames = "chr1", ranges = IRanges(start = 10, end = 20))
normalizeToMatrix(signal, target, extend = 10, w = 2)
normalizeToMatrix(signal, target, extend = 10, w = 2, include_target = TRUE)
normalizeToMatrix(signal, target, extend = 10, w = 2, value_column = "score")
```

---

print.normalizedMatrix

Print normalized matrix

Description

Print normalized matrix

Usage

```
## S3 method for class 'normalizedMatrix'
print(x, ...)
```

Arguments

x	the normalized matrix returned by normalizeToMatrix
...	other arguments

Value

No value is returned.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
# There is no example
NULL
```

---

rbind.normalizedMatrix

Bind matrix by rows

Description

Bind matrix by rows

Usage

```
## S3 method for class 'normalizedMatrix'
rbind(..., deparse.level = 1)
```

Arguments

...	matrices
deparse.level	-deparse.level

Value

A normalizedMatrix class object.

Author(s)

z.gu@dkfz.de

Examples

```
# There is no example
NULL
```

show-dispatch *Method dispatch page for show*

Description

Method dispatch page for show.

Dispatch

show can be dispatched on following classes:

- show,EnrichedHeatmapList-method, EnrichedHeatmapList-class class method
- show,EnrichedHeatmap-method, EnrichedHeatmap-class class method

Examples

```
# no example
NULL
```

show-EnrichedHeatmap-method *Draw the single heatmap with default parameters*

Description

Draw the single heatmap with default parameters

Usage

```
## S4 method for signature 'EnrichedHeatmap'
show(object)
```

Arguments

object	an EnrichedHeatmap-class object.

Details

Actually it calls draw,EnrichedHeatmap-method, but only with default parameters. If users want to customize the heatmap, they can pass parameters directly to draw,EnrichedHeatmap-method.

Value

An EnrichedHeatmapList-class object.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
# see documentation of EnrichedHeatmap
NULL
```

`show-EnrichedHeatmapList-method` *Draw a list of heatmaps with default parameters*

Description

Draw a list of heatmaps with default parameters

Usage

```
## S4 method for signature 'EnrichedHeatmapList'
show(object)
```

Arguments

`object`	an `EnrichedHeatmapList-class` object.

Details

Actually it calls `draw,EnrichedHeatmapList-method`, but only with default parameters. If users want to customize the heatmap, they can pass parameters directly to `draw,EnrichedHeatmapList-method`.

Value

An `EnrichedHeatmapList-class` object.

Examples

```
# see documentation of EnrichedHeatmap
NULL
```

`[.normalizedMatrix` *Subset normalized matrix by rows*

Description

Subset normalized matrix by rows

Usage

```
## S3 method for class 'normalizedMatrix'
x[i, j, drop = FALSE]
```

Arguments

x	the normalized matrix returned by normalizeToMatrix
i	row index
j	column index
drop	whether drop the dimension

Value

A normalizedMatrix class object.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
# There is no example
NULL
```

Index

+.AdditiveUnit, 2
[.normalizedMatrix, 22

anno_enriched, 3, *10*

copyAttr, 4

default_smooth_fun, 5, *18*
discretize, 6
dist, *7*
dist_by_closeness, 7
draw, *8*, *9*
draw *(*draw-dispatch*)*, 7
draw,EnrichedHeatmap-method
*(*draw-EnrichedHeatmap-method*)*, 8
draw,EnrichedHeatmapList-method
*(*draw-EnrichedHeatmapList-method*)*, 9
draw-dispatch, 7
draw-EnrichedHeatmap-method, 8
draw-EnrichedHeatmapList-method, 9

enriched_score, *10*, 13, *14*
EnrichedHeatmap, *4*, 10, *11*, *14*, *16*, *18*
EnrichedHeatmap-class, 11
EnrichedHeatmapList, *9*, 12
EnrichedHeatmapList-class, 13
extract_anno_enriched, 14

getSignalsFromList, *15*, 15
grid.newpage, *14*

Heatmap, *10*, *11*

locfit, *5*
loess, *5*

make_layout, *9*
makeWindows, 16

normalizeToMatrix, *4*, *6*, *10*, *13*, *15*, *16*, 17, *18*, *20*, *23*

print.normalizedMatrix, 19

rbind.normalizedMatrix, 20

show *(*show-dispatch*)*, 21
show,EnrichedHeatmap-method
*(*show-EnrichedHeatmap-method*)*, 21
show,EnrichedHeatmapList-method
*(*show-EnrichedHeatmapList-method*)*, 22
show-dispatch, 21
show-EnrichedHeatmap-method, 21
show-EnrichedHeatmapList-method, 22
stop, *18*